Association Rule Learning

From Frequent Patterns to Business Insights

Ahmed BADI

Mathematics & Machine Learning Enthusiast

ahmedbadi905@gmail.com
linkedin.com/in/badi-ahmed

2025

Outline

Introduction

Motivation

- Data often comes as **transactions** or **baskets**:
  - Shopping carts, web sessions, app clicks, medical events.
- Goal: discover patterns of the form
  $$X \Rightarrow Y$$
  meaning "when X occurs, Y tends to occur as well" [1, 4].
- Classic use case: **market basket analysis** in retail.

Market Basket Example

- Suppose many customers buying {Bread, Butter} also buy {Milk}.
- Association rule:

$$\{\text{Bread}, \text{Butter}\} \Rightarrow \{\text{Milk}\}$$

- Applications:
  - Product placement and promotions.
  - Recommendation and cross-selling.

Key Concepts

Basic Definitions

- Item set:
$$\mathcal{I} = \{i_1, i_2, \ldots, i_m\}$$
- Dataset of transactions:
$$\mathcal{D} = \{T_1, \ldots, T_N\}, \quad T_j \subseteq \mathcal{I}$$
- **Itemset** X: any subset $X \subseteq \mathcal{I}$.
- Size of X: its cardinality $|X|$ (2-itemset, 3-itemset, etc.).

Association Rule

Definition

An association rule is an implication

$$X \Rightarrow Y$$

where $X, Y \subseteq \mathcal{I}$, $X \cap Y = \emptyset$, $X \neq \emptyset$, $Y \neq \emptyset$.

- X: **antecedent** (left-hand side).
- Y: **consequent** (right-hand side) [4].
- Objective: find rules that are both **frequent** and **strong**.

Metrics: Support, Confidence, Lift

Support

Support of an Itemset

$$\text{supp\_count}(X) = |\{T_j \in \mathcal{D} : X \subseteq T_j\}|$$

$$\text{supp}(X) = \frac{\text{supp\_count}(X)}{N}$$

- Measures how **frequently** X occurs in the dataset.
- For a rule $X \Rightarrow Y$, support is $\text{supp}(X \cup Y)$.

Confidence

Confidence of a Rule

$$\text{conf}(X \Rightarrow Y) = \frac{\text{supp}(X \cup Y)}{\text{supp}(X)} = P(Y \mid X)$$

- Measures how often Y appears in transactions containing X.
- Directional: $\text{conf}(X \Rightarrow Y) \neq \text{conf}(Y \Rightarrow X)$ [4].

Lift

Lift of a Rule

$$\mathrm{lift}(X \Rightarrow Y) = \frac{\mathrm{conf}(X \Rightarrow Y)}{\mathrm{supp}(Y)} = \frac{\mathrm{supp}(X \cup Y)}{\mathrm{supp}(X)\,\mathrm{supp}(Y)}$$

- Compares observed co-occurrence to what is expected under independence [4].
- Interpretation:
  - > 1: positive association.
  - $= 1$: independence.
  - < 1: negative association.

Example Rules

Rule	Support	Confidence	Lift
Bread $\Rightarrow$ Butter	0.20	0.50	1.25
Butter $\Rightarrow$ Bread	0.20	1.00	2.50
Bread $\Rightarrow$ Milk	0.30	0.75	1.50

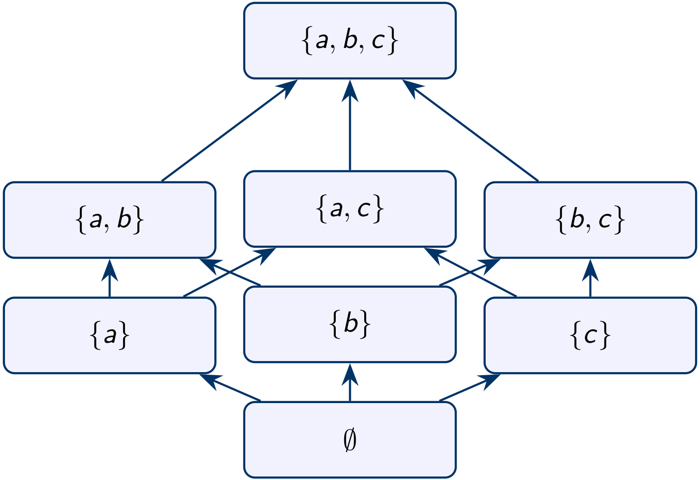
Table: Illustrative rule metrics.

Frequent Itemset Mining

From Itemsets to Rules

- Most algorithms operate in two phases [1, 4]:
  1. **Frequent itemset mining**: find all X with $\text{supp}(X) \geq \text{min\_supp}$.
  2. **Rule generation**: from each frequent itemset L, generate rules $X \Rightarrow L \setminus X$ meeting min_conf (and possibly lift).
- Challenge: there are 2^m possible itemsets.

Search Space



Lattice of itemsets; algorithms must explore or prune this space.

Apriori Algorithm

Apriori Property

Key Property [1, 4]

If an itemset is **frequent**, all of its subsets are frequent.

If an itemset is **not** frequent, no superset can be frequent.

- Anti-monotonicity of support.
- Enables strong pruning of the search space.

Apriori: Level-wise Procedure

1. Choose minimum support min_supp.
2. Find frequent 1-itemsets L_1.
3. For $k = 2, 3, \ldots$:
   - Generate candidate k-itemsets C_k from L_{k-1}.
   - Prune candidates whose $(k-1)$-subsets are not all in L_{k-1}.
   - Scan database, compute support of candidates.
   - Form $L_k = \{c \in C_k \mid \text{supp}(c) \geq \text{min\_supp}\}$.
4. Stop when L_k becomes empty; output $\bigcup_k L_k$.

Apriori: Toy Illustration

- Start with frequent 1-itemsets: $\{Bread\}$, $\{Butter\}$, $\{Milk\}$.
- Generate candidates:
  - 2-itemsets: $\{Bread, Butter\}$, $\{Bread, Milk\}$, $\{Butter, Milk\}$.
  - Keep only those above min_supp.
- Continue to 3-itemsets if any 2-itemsets are frequent.
- Stop when no larger frequent itemsets exist.

FP-Growth & Eclat

FP-Growth: Motivation

- Apriori may generate many candidates and scan the database multiple times.
- FP-Growth avoids explicit candidate generation using an **FP-tree** [2, 4].
- Typically needs only two passes over the data.

FP-Growth: Main Steps

1. First pass: compute frequent items and their supports.
2. Reorder items in each transaction by descending support, discard infrequent items.
3. Build FP-tree: insert each ordered transaction, sharing common prefixes.
4. Recursively mine the FP-tree:
    - For each item, build a conditional pattern base.
    - Construct conditional FP-tree and mine for patterns containing that item.

Eclat: Vertical Representation

- Eclat stores data in **vertical** form: TIDLists [3].
- For each item i:

$$\mathrm{TIDList}(i) = \{j \mid i \in T_j\}$$

- For itemset $X = \{i_1, \ldots, i_k\}$:

$$\mathrm{TIDList}(X) = \bigcap_{\ell=1}^{k} \mathrm{TIDList}(i_\ell)$$

- Support count is $|\mathrm{TIDList}(X)|$.

Eclat: Depth-First Mining

- Start from frequent 1-itemsets with their TIDLists.
- Extend an itemset X by intersecting its TIDList with those of following items.
- Depth-first exploration of the itemset lattice.
- Efficient when TIDLists are relatively small (sparse data).

Algorithm Comparison

Algorithm	Search Strategy	Structure	Key Strength
Apriori	Level-wise, breadth-first	None (flat)	Simple; strong pruning with Apriori property
FP-Growth	Pattern growth, divide-and-conquer	FP-tree	Few scans; efficient on dense data
Eclat	Depth-first	TIDLists (vertical)	Fast intersections on sparse data

Table: Frequent itemset mining algorithms [1, 2, 3].

Rule Generation & Applications

From Frequent Itemsets to Rules

- For each frequent itemset L ($|L| \geq 2$) [1, 4]:
  1. Generate all non-empty proper subsets $X \subset L$.
  2. For each X, form rule $X \Rightarrow L \setminus X$.
  3. Compute
  $$\text{conf}(X \Rightarrow Y) = \frac{\text{supp}(L)}{\text{supp}(X)}.$$
  4. Keep rules with confidence (and optionally lift) above thresholds.
- Post-processing: remove redundant or uninteresting rules.

Applications

- **E-commerce & retail**:
  - Market basket analysis, cross-selling, product placement.
- **Food delivery platforms**:
  - Suggesting combos (burger + fries, pizza + drink).
- **Streaming services**:
  - Co-consumption patterns across movies, series, or songs.
- **Finance, healthcare, fitness**:
  - Spending patterns, co-occurring diagnoses or treatments [4].

Practical Considerations

- **Threshold tuning**:
  - Low thresholds $\Rightarrow$ too many, noisy rules.
  - High thresholds $\Rightarrow$ missed interesting patterns.
- **Redundancy**:
  - Many rules may convey similar information.
  - Need ranking and pruning.
- **Domain validation**:
  - Statistical significance $\neq$ business value.
  - Expert review and A/B testing often required [4].

Conclusion

Conclusion

- Association rule learning discovers interpretable co-occurrence patterns in transactional data.
- Metrics: **support**, **confidence**, **lift** quantify rule quality.
- Algorithms Apriori, FP-Growth, and Eclat offer different trade-offs for frequent itemset mining [1, 2, 3, 4].
- With appropriate thresholds and domain expertise, association rules power recommendation, marketing analytics, and exploratory data mining.

References

R. Agrawal and R. Srikant,
"Fast Algorithms for Mining Association Rules,"
VLDB, 1994.

J. Han, J. Pei, and Y. Yin,
"Mining Frequent Patterns without Candidate Generation,"
SIGMOD, 2000.

M. J. Zaki,
"Scalable Algorithms for Association Mining,"
IEEE TKDE, 12(3), 2000.

J. Han, M. Kamber, and J. Pei,
Data Mining: Concepts and Techniques,
Morgan Kaufmann, 2011.